CDAP: A Cultural Algorithm for Data Placement in Big Data Workflows

Mahdi Ebrahimi
Computer Science Department
CSU Northridge
Northridge, United States
mahdi.ebrahimi@csun.edu

Seyed Ziae Mousavi Mojab
College of Business & IT
Lawrence Tech. University
Southfield, Michigan, United States
smousavim@ltu.edu

Najmeh Najmi
Computer Science Department
CSU Northridge
Northridge, United States
najmeh.najmi.796@my.csun.edu

Abstract— **The performance of executing big data workflows in the Cloud highly depends on the placement of the workflow original datasets. An optimal data placement reduces the data movement among virtual machines, and as a result, it significantly reduces the workflow makespan. Data placement is an NP-hard problem; therefore in this paper, we propose CDAP (Cultural algorithm Data Placement), a novel metaheuristic data placement strategy based on Cultural Algorithms (CA) to improve the performance of a workflow by minimizing the data movement among the virtual machines during workflow executions. The effectiveness of CDAP is demonstrated through extensive experiments where we evaluated our proposed method against a set of well-known data placement strategies.**

Keywords—Big Data, Big Data Workflows, Data Placement, Cloud Computing, Cultural Algorithms
Full/Regular Research Paper (CSCI-RTBD)

I. INTRODUCTION

The advances in computing and data storage technologies brings the opportunity to scientists to perform more collaborative research. Workflows are generally used to model complex scientific computations that bring an opportunity to scientists to effectively and efficiently collaborate by sharing and reusing their workflows and, therefore, reproduce their experimental results. Big data workflows naturally include hundreds and thousands of linked computation tasks to process a large number of scientific datasets. It is feasible to execute big data workflows over distributed, and heterogeneous computing environments such as cloud [7, 10].

A big data workflow management system is a platform to design and execute complex workflows and traditionally is modeled by a directed acyclic graph (DAG) where each graph node represents a workflow computation task, and the directed edges between two task nodes represent dataflow between tasks.

In this paper, we propose CDAP, an evolutionary algorithm (EA) that is based on Cultural Algorithms (CA) [9] to optimally apply the data placement for big data workflow executions in the Cloud. As big data workflows are data-centric applications and process huge datasets, therefore, our main goal is to minimize the total data movement between cloud virtual machines during the workflow executions. A sample workflow with five tasks (t_1 – t_5), and five datasets (d_1 – d_5) is shown in Figure 1.a. Figure 1.b shows an instance of the tasks and datasets placement for the sample workflow in the cloud with three virtual machines (VM_1 – VM_3).

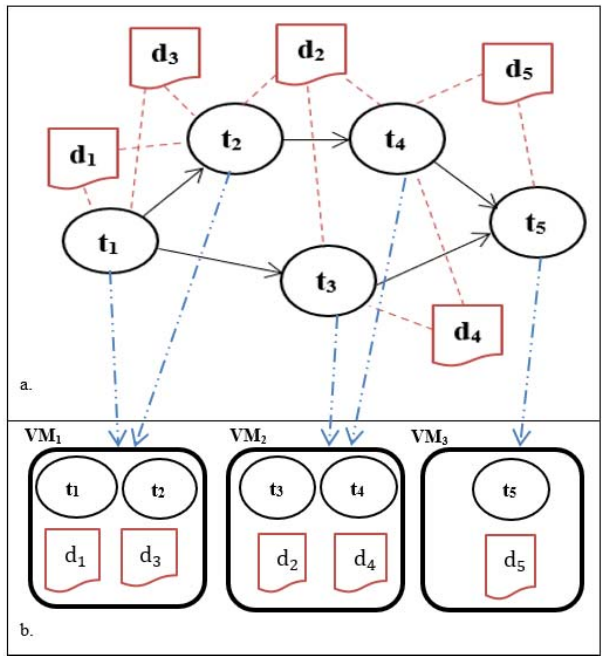

Figure 1. a) a workflow with five tasks $\{t_1, t_2, \dots, t_5\}$ and five datasets $\{d_1, d_2, \dots, d_5\}$. $\{d_1, d_3\}$ are the input datasets for task t_1, $\{d_1, d_2, d_3\}$ are input datasets for task t_2, and so on. b). a cloud virtual machine setting with three virtual machines. Datasets $\{d_1, d_3\}$ as well as tasks $\{t_1, t_2\}$ are placed in VM_1. Datasets $\{d_2, d_4\}$, and tasks $\{t_3, t_4\}$ are placed in VM_2. Similarly, dataset $\{d_5\}$ as well as task $\{t_5\}$ are placed in VM_3.

Here, VM_1 hosted tasks t_1 and t_2 as well as datasets, d_1 and d_3. Similarly, VM_2 hosted two tasks t_3 and t_4 along with two datasets, d_2 and d_4. Additionally, both task t_5 and dataset d_5 are placed in VM_3. Once we execute this workflow, dataset d_2 is required to move from VM_2 to VM_1 to complete the process of task t_2. However, there is no need to move any other datasets from other virtual machines into VM_1 to run task t_1 because all its input datasets, d_1, and d_3 are already placed in VM_1.

In this paper we propose a CA-based data placement algorithm named CDAP to minimize the total data movement between cloud virtual machines during workflow executions. CDAP randomly generates a list of data placement solutions. Then within some iterations, CDAP computes and compares the generated data placement solutions by applying a pre-defined fitness score and at the end, it returns the best data placement solution. The best data placement solution is the one that minimizes the total data movement between cloud virtual machines during workflow executions.

In this paper, we only focus on the data placement problem, and the workflow scheduling problem is out of the scope of this work. Any workflow scheduling algorithm can utilize CDAP to improve the workflow throughput. To evaluate CDAP and compare it with the most competitive data placement algorithms, we execute workflow tasks sequentially using topological sort. We explain our proposed strategy for data placement in section III in detail.

The remainder of this paper is organized as follows: in section II, we define and formalize our system model. Section III explains our data placement strategy, CDAP, in detail. Then in section IV, the experimental results are shown and discussed. Section V presents the related work. Finally, the conclusion and future work are presented in section VI.

II. FORMALIZING WORKFLOW DATA PLACEMENT PROBLEM

Big data workflows are executed in the Cloud. Therefore, we model the cloud computing environment first. We consider *M* distributed virtual machines in the Cloud as the execution sites to execute workflow tasks. Cloud Computing Providers (CCP) typically have their own data-balancing strategy to store data and assign computation tasks to proper virtual machines; they do not consider the structure of the workflows.

To execute big data workflows in the Cloud, we need to model the Cloud. A cloud computing environment is modeled as follows:

Definition 2.1 (Cloud Computing Environment *C*). A Cloud computing environment C is a 3-tuple $C = (VM, VMS, DTR)$, where:

- VM is a set of virtual machines in the cloud *C*. VM_m denotes the m^{th} virtual machine in *C*.
- *VMS:* $VM \rightarrow R^+$ is the virtual machine storage capacity function. $VMS(VM_m)$ returns the maximum available storage capacity of the virtual machine VM_m in the cloud *C*. It is measured in some pre-determined units such as Gigabyte (GB) or Terabyte (TB). R^+ is the set of all real positive numbers.
- $DTR: VM \times VM \rightarrow R^+$ is the data transfer rate function. $DTR(VM_{m1}, VM_{m2})$ returns the network bandwidth between two virtual machines VM_{m1} and VM_{m2}. It is measured in some pre-determined unit, such as Megabyte (MB) per second. R^+ is the set of all real positive numbers.

In this paper we assume that all the cloud virtual machines are used for both computation and storage purposes. Big data workflows contain a set of computation tasks that consume one or more datasets and may produce intermediate datasets as their outputs. Those output datasets will be sent to other tasks as their inputs by following the data flow logic, represented as edges in workflow graphs. A big data workflow is formalized as follows:

Definition 2.2 (Big Data Workflow *W*). A big data workflow *W* can be modeled formally as a 4-tuple including two sets and two functions as follows:

$$W = (T, D, DSize, DS)$$

- T is the set of workflow computation tasks. Each individual task is denoted by t_k.
- D is the set of input datasets for workflow *W*. Each individual dataset is denoted by d_i.
- $DSize: D \rightarrow R^+$ is the dataset size function. $DSize(d_i)$ returns the size of the dataset d_i. The size of a dataset is defined in some pre-determined units such GB.
- $DS: T \rightarrow D$ is the task-dataset function. $DS(t_k)$, $t_k \in T$ returns the set of datasets that are consumed by t_k as its inputs.

Definition 2.3 (Data Movement Cost *DMC*). To access and transfer the dataset d_i from virtual machine VM_{m1} to VM_{m2}, we calculate the data movement cost (*DMC*) by:

$$DMC(d_i, VM_{m1}, VM_{m2}) = \begin{cases} 0, & if\ m_1 = m_2 \\ \dfrac{DSize(d_i)}{DTR(VM_{m1}, VM_{m2})}, & if\ m_1 \neq m_2 \end{cases}$$

Data placement scheme is defined to represent the place of each workflow dataset in a virtual machine and is defined formally as follows:

Definition 2.4 (Data Placement Scheme *P*). Suppose there are *M* virtual machines and *I* datasets, a data placement scheme is represented by a *I*-element vector P such that $P(d_i)$ indicates the index of the virtual machine which d_i is placed into. For example, the data placement scheme of the example workflow in Figure 1.a is $\mathrm{P} = (1, 2, 1, 2, 3)$, and it means that the datasets d_1 and d_3 are placed into the virtual machine VM_1 $(P(d_1) = P(d_3) = VM_1)$, the datasets d_2 and d_4 are placed into the virtual machine VM_2 $(P(d_2) = P(d_4) = VM_2)$ and the dataset d_5 is placed to the virtual machine VM_3 $(P(d_5) = VM_3)$.

After data placement, CDAP assigns workflow tasks to the most appropriate virtual machine. $Assign(t_k)$ function is used to indicate the virtual machine that t_k is assigned to. For example in Figure 1.a, $Assign(t_1) = Assign(t_2) = VM_1$, $Assign(t_3) = Assign(t_4) = VM_2$,and $Assign(t_5) = VM_3$.

To evaluate and compare CDAP with other competitive, existing algorithms, we define the fitness score as the total size of data movement during workflow executions. For this purpose, we execute all the K number of workflow tasks, and for each task execution, we summate the entire input datasets movement cost using *DMC* definition (Definition 2.3). The fitness score is defined as follows:

Definition 2.5 (Fitness Score *FScore*). Given a workflow *W* with *K* number of tasks executing in the cloud *C* with utilizing the data placement *P*, fitness score is equal to the total data movement for executing all tasks of the workflow *W*, in the cloud C. *FScore* is defines as follows:

$$FScore(W, C, P) = \sum_{k=1}^{K} \sum_{\substack{d_i \in DS(t_k) \\ t_k \in T}} DMC\big(d_i, P(d_i), Assign(t_k)\big)$$

For the data placement problem addressed in this paper, each virtual machine has limited storage capacity and can store multiple datasets subject to its storage capacity constraint.

The goal is to place the workflow datasets into the cloud virtual machines such that the *FScore* gets minimized. Therefore, we define the data placement minimization problem as follows:

Definition 2.6 (Data Placement Minimization P_{opt}). Given a workflow *W,* a cloud computing environment *C,* and data placement schema *P,* the data placement problem is formalized to search for the optimal data placement schema, P_{opt} as:

$$P_{opt} = \underset{P}{argmin}\ \ FScore(W, C, P)$$

In the next section, we present CDAP approach to search for the optimal data placement solution, P_{opt} in the space of all possible data placement solutions.

III. PROPOSED DATA PLACEMENT ALGORITHM

CDAP is an Evolutionary Algorithms (EA) based on the sCultural Algorithms (CA) that simulates cultural evolution [9].

CA is a dual-inheritance system that has two main components: The Population Space and the Belief Space. The Belief Space is an additional component that provides a mechanism to store and transfer knowledge from one generation to another generation. Moreover, both Population and Belief Spaces can communicate with each other to exchange their knowledge using acceptance and influence protocols. (Figure 2).

Any traditional heuristic algorithms like Genetic Algorithm (GA) can be employed in the Population Space. To explore the search space and generate the optimal solution, genetic operators such as selection, mutation, and crossover can be applied.

The Belief Space component can include various knowledge sources. Three primary sources utilized in CDAP are as follows:

1) Domain knowledge that tracks changes in the fitness scores to generate diversity and mutation values. 2) Situational Knowledge that comprises the best solutions (Elites) from the population, and 3) Normative knowledge that contains the upper and lower bounds for various numeric attributes like virtual machine or dataset indices.

After producing each generation, an acceptance function is applied to select the elite solutions to update the knowledge sources in the Belief Space. The Belief Space can influence the Population Space by guiding the changes in the data placement solutions for the next generation [9]. Population and Belief Spaces are updated after each generation based on feedback from each other. These processes repeat until reaching pre-specified termination conditions.

The main steps of the CDAP algorithm are shown in Figure 3. Algorithm 1 presents the pseudo-code for the CDAP algorithm. CDAP has three main inputs: Workflow W, cloud virtual machines specifications VM, and CDAP configurations Config. In the first step, CDAP randomly creates the initial population (the data placement solutions) and initializes the Belief Space (lines 1-3). Then, it evaluates the performance of each individual solution using the fitness score (line 4). In the next step, CDAP selects the best individuals (elites) to update the Belief Space (lines 6-7).

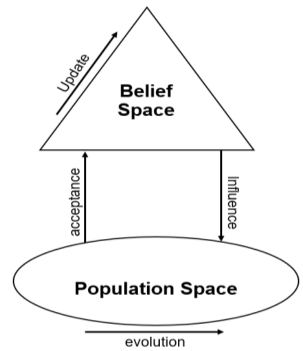

Figure 2: Cultural Algorithms

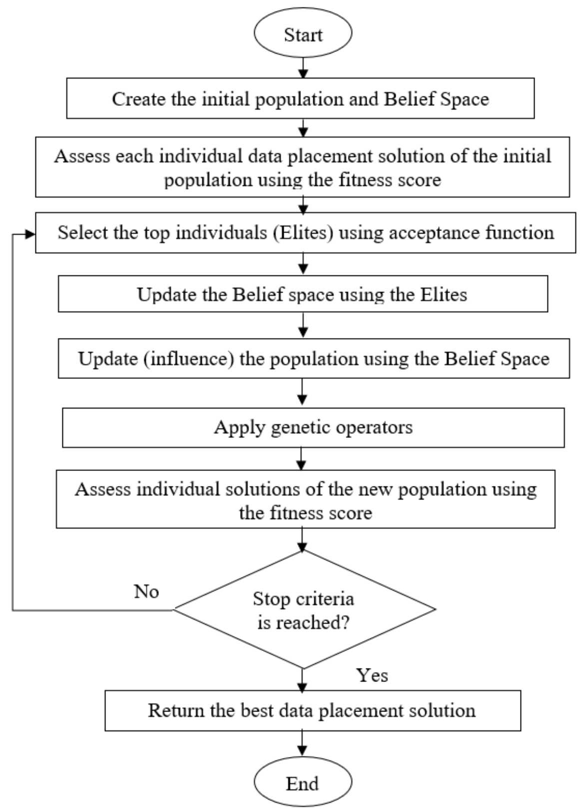

Figure 3. Flowchart of CDAP

Algorithm 1: CDAP

Input: Workflow *W*, Virtual machines specifications *VM*, CDAP configurations

Output: The optimal data placement solution P_{opt}

1. t ← 0
2. Initialize Population POP(t);
3. Initialize Belief Space BLF(t);
4. Evaluate Population FScore(POP(t), Obj());
5. **Repeat**
6. Accept(POP(t));
7. Update(BLF(t), Accept(POP(t)));
8. POP(t+1) ← Influence(POP(t), BLF(t));
9. Variation(POP(t+1));
10. t ← t+1;
11. **Until** termination condition achieved
12. Return the best data placement solution, P_{opt}

Then, CDAP influences the current population based on the knowledge in the Belief Space by moving them towards areas of highest fitness, and generates a new population evolved from the old population (lines 8-9). Next, CDAP returns the best solution if the stop criteria are reached (line 12), otherwise it will repeat lines 6-10.

IV. EXPERIMENTAL RESULTS

The experimental results are presented and discussed in this section. We compare CDAP with the most competitive data placement approaches, PSO, GA and Random algorithms.

A. Performance Evaluation

To evaluate the performance of CDAP, we adopted four synthetic workflow applications based on real scientific workflows: Montage (astronomy), CyberShake (earthquake science),Epigenomics (biogenetics), and LIGO (gravitational physics domain) [3]. Figure 4 shows the structure of these workflows. We compare different sizes of these workflow applications and assume each of the workflow task can be run on every cloud virtual machine.

We demonstrate the performance of CDAP and other approaches regarding the average of the total data movement cost in terms of hour. In our experiments, we assume that all the virtual machine types are the same and the data movement rates (network bandwidth) between all virtual machines are the same. Table 1 shows the value of the parameters used in our experiments.

B. Results And Analysis

Figure 5 shows the data movement cost (*DMC*), in terms of hour by varying the size of workflows. We use five virtual machines. CDAP reduces DMC and outperforms the other algorithms. In addition, by increasing the size of the workflows, *DMC* is increased in all algorithms. However, CDAP has a better performance compared to the other algorithm in more complex, large workflows.

In Figure 6, we compare CDAP with the other algorithms by increasing the number of cloud virtual machines. We use three, five, ten, and fifteen virtual machines for each workflow of size 100. We assume all the virtual machines are the same type. *DMC* is increased by increasing the number of virtual machines in all four strategies as there are more options (virtual machine) for the data placement algorithms to place the datasets. Again, CDAP outperforms the other strategies and reduces DMC more than the other algorithms. There is a higher performance improvement using CDAP in a larger number of virtual machines.

The experimental results prove that CDAP effectively decreases workflow communication costs more than PSO, GA, and Random approaches.

V. RELATED WORK

Both data placement and workflow task assignment have become fundamental research topics in the Cloud due to the rapid increase of accessible large datasets over the Internet and the emerging field of big data [9]. Current research studies for cloud computing environments have been mainly focused on the optimization of task scheduling and data placement. In [11], the authors considered data placement with data replicas for distributed environments. They grouped the most similar data together based on their occurrences in

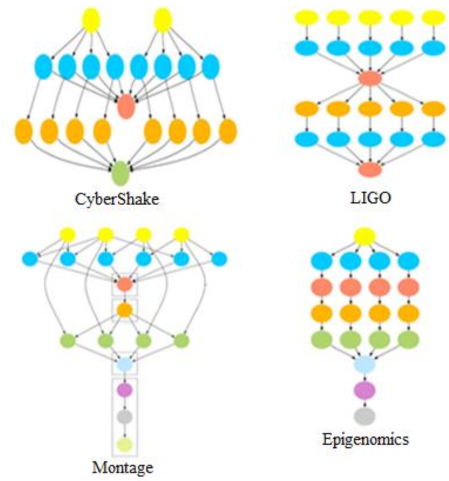

Figure 4: Scientific workflows used in the experiments [3].

TABLE 1: DEFAULT SETTINGS USED IN THE EXPERIMENTS.	
Size of Population	100 - 300
Maximum generation	100
Crossover probability	0.85 - 0.95
Mutation probability	0.10 - 0.20
Maximum iteration	20
Number of Elites	0.15 * Pop

common query accesses to minimize the average number of dedicated computation nodes. In [6], an Ant-Colony data placement algorithm is proposed such that each ant places datasets in the proper data center based on the heuristic and pheromone information. By applying task placement and data replication services, paper [5] evaluated and displayed the benefits of pre-staging data compared to the Pegasus data stage processing. A Genetic Algorithm for data placement was proposed in [8]. This paper considered a load-balancing factor to reduce data movement, but the workflow structure was not considered. The particle swarm optimization (PSO) was utilized for data placement in [2, 4]. In addition, in paper [1], the authors proposed a data placement algorithm by combining both GA and PSO algorithms.

In our previous works, we proposed big data placement strategy (BDAP) [12] and task placement (TPS) [14] in order to place the most interdependent datasets and tasks in the same cloud virtual machine. BDAP minimized the total amount of data movement between virtual machines during workflows executions. TPS is about task assignment and can be applied independently or conjectured with BDAP to place data and workflow takes together. All these works, including BDAP and TPS are either GA or PSO-based approaches, and our main contribution in this paper is to improve the performance of the data placement algorithm by employing CA.

VI. CONCLUSIONS AND FUTURE WORK

We propose CDAP, a data placement strategy for Cloud-based scientific workflow by utilizing Cultural Algorithms. CDAP minimizes the total amount of data movement between virtual machines during executions of the workflows in the Cloud. In this work, we employed the Cultural Algorithm (CA) to select and place the most proper dataset in the same

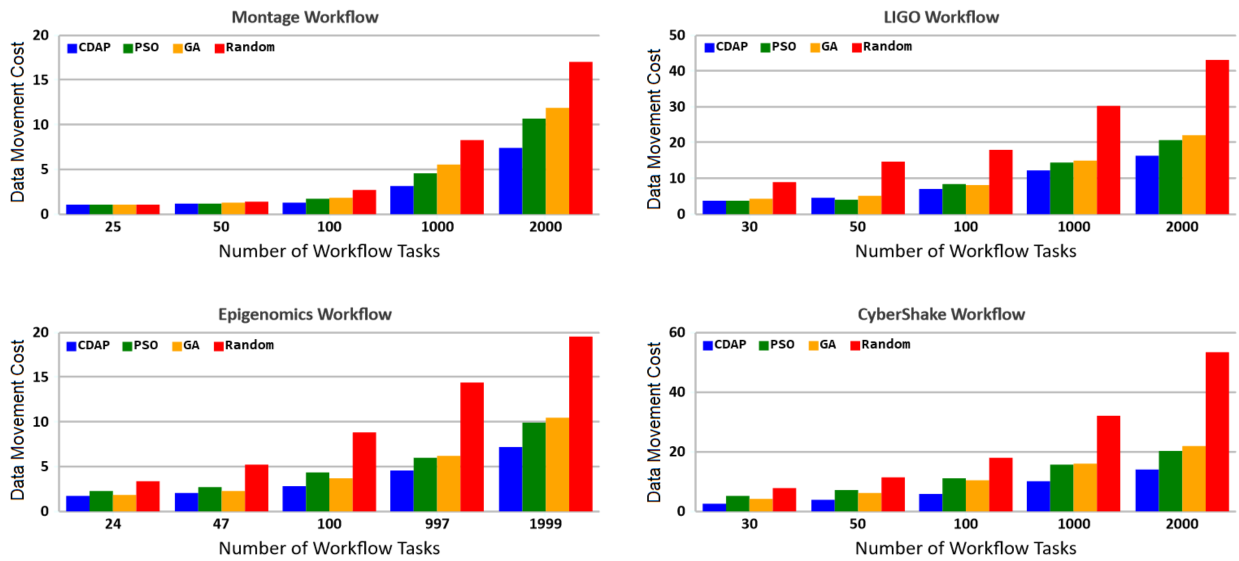

Figure 5. Data Movement (DMC) Cost by increasing the size of workflows.

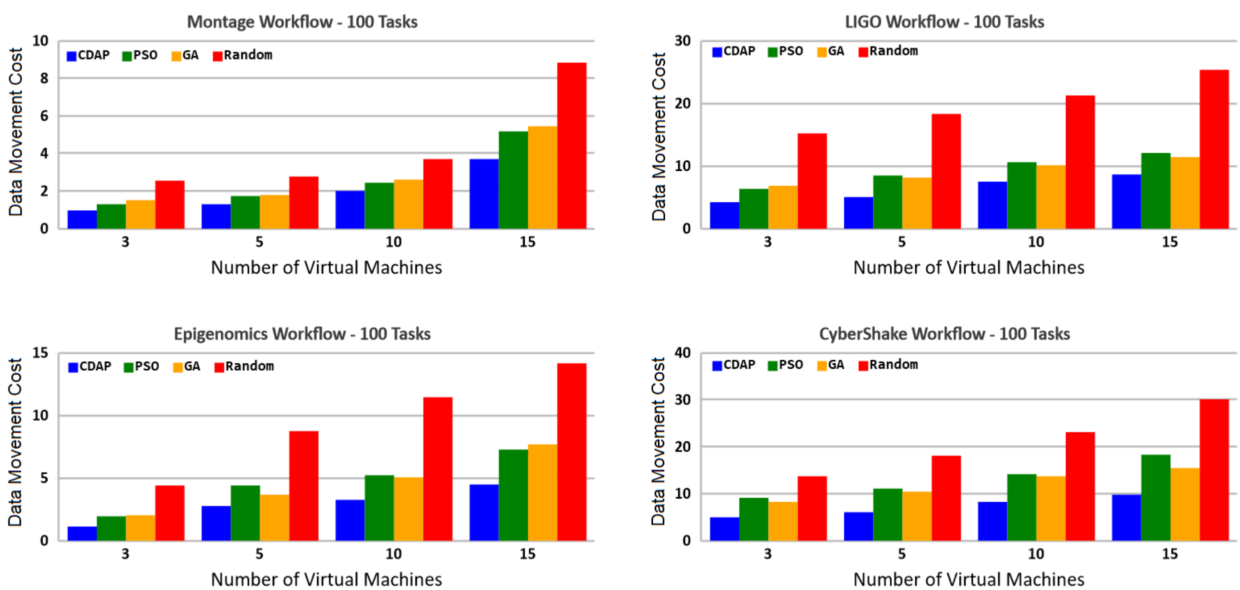

Figure 6. Data Movement Cost (DMC) by increasing the number of cloud virtual machines.

virtual machine during workflow executions. Our extensive experiments and comparisons have shown that CDAP outperforms other proposed algorithms in minimizing data movement.

For future work, we plan to consider both data and task replica to improve CDAP. In addition, we plan to extend CDAP to achieve data placement for the execution of various workflows simultaneously.

REFERENCES

[1] Z. Chen, J. Hu, G. Min, and X. Chen, "Effective data placement for scientific workflows in mobile edge computing using genetic particle swarm optimization," in Proceedings of the Concurrency and Computation: Practice and Experience, vol. 33, 2019.

[2] Z. Liu, T. Xiang, B. Lin, X. Ye, H. Wang, Y. Zhang, and X. Chen., "A Data Placement Strategy for Scientific Workflow in Hybrid Cloud," in Proceedings of the IEEE 11th International Conference on Cloud Computing (CLOUD), pp. 556-563, 2018.

[3] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M. Su and K. Vahi "Characterization of Scientific Workflows," in Proceedings of the 3rd Workshop on Workflows in Support of Large-Scale Science (WORKS), 2008.

[4] X. Du, S. Tang, Z. Lu, J. Wet, K. Gai and P. C. K. Hung, "A Novel Data Placement Strategy for Data-Sharing Scientific Workflows in Heterogeneous Edge-Cloud Computing Environments," in Proceedings of the IEEE International Conference on Web Services (ICWS), pp. 498-507, 2020.

[5] A. Chervenak, E. Deelman, M. Livny, M. Su, R. Schuler, S. Bharathi, G. Mehta, and K. Vahi, "Data Placement for Scientific Applications in Distributed Environments". in Proceedings of Grid Conference, Austin, Texas, 2015.

[6] L. Bing, F. Zhu, J. Zhang, J. Chen, X. Chen, N. Xiong and J. Mauri. “A Time-Driven Data Placement Strategy for a Scientific Workflow Combining Edge Computing and Cloud Computing.” IEEE Transactions on Industrial Informatics 15: 4254-4265, 2019.

[7] Y. Zhao, I. Raicu, X. Fei and S. Lu, "Opportunities and Challenges in Running Scientific Workflows on the Cloud". International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011.

[8] Z. Er-Dun, Q. Yong-Qiang, X. Xing-Xing and C. Yi "A Data Placement Strategy Based on Genetic Algorithm for Scientific Workflows". Proceedings of the Eighth International Conference on Computational Intelligence and Security (CIS), 2012.

[9] S. Z. Mousavi Mojab, M. Ebrahimi, R. Reynolds and S. Lu, "iCATS: Scheduling Big Data Workflows in the Cloud Using Cultural Algorithms," in Proceedings of the Fifth IEEE International Conference on Big Data Computing Service and Applications (BigDataService), pp. 99-106, 2019.

[10] G. Juve and E. Deelman, "Scientific workflows and clouds". Crossroads, vol. 16, pp. 14-18, 2017.

[11] A. K. Kayyoor, A. Deshpande and S. Khuller, "Data Placement and Replica Selection for Improving Co-location in Distributed Environments". CoRR abs/1302.4168, 2018.

[12] M. Ebrahimi, A. Mohan, A. Kashlev, and S. Lu, “BDAP: A Big Data Placement Strategy for Cloud-Based Scientific Workflows”, in Proceedings of the First IEEE International Conference on Big Data Computing Services and Applications, pp. 105-114, 2015.

[13] D. Yuri, P. Membrey, P. Grosso and C. Laat "Addressing Big Data Issues in Scientific Data Infrastructure". The first International Symposium on Big Data and Data Analytics in Collaboration (BDDAC). Part of the International Conference on Collaboration Technologies and Systems (CTS), San Diego, California, USA, 2018.

[14] M. Ebrahimi, A. Mohan, S. Lu and R. Reynolds, "TPS: A task placement strategy for big data workflows," in Proceedings of the IEEE International Conference on Big Data (Big Data), pp. 523-530, 2015.